WL-TR-92-3100

ASTROS-ID:
SOFTWARE FOR SYSTEM IDENTIFICATION USING MATHEMATICAL PROGRAMMING

Warren C. Gibson
CSA Engineering, Inc.
2580 W. Bayshore Road
Palo Alto, CA 94303

September 1992

Final Report for Period June 1991 - December 1991

Approved for public release; distribution unlimited.

Flight Dynamics Directorate
Wright Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, Ohio 45433-6553

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise as in any manner, as licensing the holder or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

STEPHEN M. RASMUSSEN, Capt, USAF
Project Engineer
Design & Analysis Methods Group

NELSON D. WOLF, Technical Manager
Design & Analysis Methods Group
Analysis & Optimization Branch

FOR THE DIVISION

DAVID K. MILLER, Lt Col, USAF
Chief, Analysis & Optimization Branch
Structures Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/FIBRA, Wright-Patterson AFB OH 45433-6553 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific doucment.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 1992	**3. REPORT TYPE AND DATES COVERED** FINAL (JUN 91 - DEC 91)

4. TITLE AND SUBTITLE

ASTROS-ID: Software for System Identification
Using Mathematical Programming

5. FUNDING NUMBERS

C-F33615-90-C-3211
PROG. ELEMENT: 62201F
PROG: 2401
TASK: 02
WU: 92

6. AUTHOR(S)

Warren C. Gibson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CSA Engineering, Inc.
2850 W. Bayshore Road
Palo Alto, CA 94303

8. PERFORMING ORGANIZATION REPORT NUMBER

ASIAC-TR-92-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Stephen Rasmussen, Capt, USAF 512-255/6992
Flight Dynamics Directorate (WL/FIBR)
Wright Laboratory
Wright Patterson AFB OH 45433-6553

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

WL-TR-92-3100

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

ASTROS is a multidisciplinary optimization computer program developed for Wright Laboratory. It seeks minimum weight structural designs subject to constraints on displacements, stresses, vibration frequencies, flutter speeds, etc. The program is based on a versatile set of software tools including a relational database, a matrix programming language, and an input description facility, all of which make the program readily extensible by knowledgable users. This report documents an extension to ASTROS (called ASTROS-ID) which makes it possible to tune finite element models to enhance their correspondence with measured frequencies and mode shapes. ASTROS existing optimization package is used to construct a constrained minimization problem where the user may choose to include particular frequencies and mode shape entries either in the objective function, in a least squares sense, or in constraints which require conformance within a given tolerance. Other terms include an excursion penalty and a weight deviation penalty. At the time of writing, ASTROS-ID was still in the research stage and had not been subject to beta testing.

14. SUBJECT TERMS Optimization, System Identification, ASTROS, Natural Frequencies, Mode Shapes

15. NUMBER OF PAGES 44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

FOREWORD

This report was prepared by the Aerospace Structures Information and Analysis Center (ASIAC), which is operated by CSA Engineering, Inc. under contract number F33615-90-C-3211 for the Flight Dynamics Directorate, Wright-Patterson Air Force Base, Ohio. The report presents the work performed under ASIAC Task No. T-10. This effort was sponsored by the Analysis and Optimization Branch, Structures Division, Flight Dynamics Directorate, WPAFB, Ohio, with Major Mark Ewing as the technical monitor. The analysis was performed by Dr. Warren C. Gibson, CSA Engineering, Inc.

Contents

1. **Introduction** 1

2. **System Identification using ASTROS** 2

2.1 Mathematical Formulation 2

2.2 Optimization Strategy 4

2.2.1 Eigenvalue Analysis 4

2.2.2 Sensitivity Analysis 5

2.2.3 Approximate Problem 5

2.2.4 Cross-orthogonality Check 6

2.3 The User Interface in ASTROS-ID 7

3. **Review of the ASTROS Software Structure** 9

3.1 MAPOL, the Matrix Language 9

3.2 CADDB, the Relational Database 9

3.3 The SYSGEN Process 10

3.3.1 Functional Module Definition 10

3.3.2 Standard Solution Algorithm Definition 10

3.3.3 Bulk Data Template 10

3.3.4 Relational Schema Definition 11

3.3.5 Error Message Text Definition 11

3.4 ICE: interactive access to CADDB 11

4. **Implementation** 12

5. **Sample Problems** 13

5.1 AMRAAM Beam Model 13

5.2 Intermediate Wing Box 16

5.2.1 Possible Local Minimum 16

5.2.2 Results using objective function terms 18

5.2.3 Results using constraints 18

6. **Conclusions and Extensions** 20

6.1 Questions Remaining to be Addressed 20

6.2 Caveats and Limitations 20

6.3 Recommendations for further work 21

7. References 22

Appendix A: Nelson's Method for Eigenvector Sensitivity Analysis in NASTRAN or ASTROS 23

Appendix B: Bulk Data Entries 27

Appendix C: List of New Modules in ASTROS-ID 32

Appendix D: List of New Database Entities in ASTROS-ID 36

List of Figures

1 AMRAAM Beam Model 13

2 Intermediate wing box 16

3 Contour plot of the "exact" objective function for the intermediate wing. 17

4 Contour plot of the approximate objective function for the intermediate wing. 17

List of Tables

1 AMRAAM beam, first run. 14

2 AMRAAM beam, second run. 14

3 AMRAAM beam, third run. 15

4 AMRAAM beam, fourth run. 15

5 Design history for the intermediate wing, using the objective function. 18

6 Design history for the intermediate wing, using constraints. 19

1. Introduction

This report documents a portion of ASIAC Task 10, "AMRAAM Missile Vibration Prediction." This part of the Task 10 effort was concerned with implementation of a system identification or model tuning technique within the framework of ASTROS. A separate report documents work that was done on updating of finite element models of the AMRAAM missile.

The end product of this part of the Task 10 effort was a new version of ASTROS, called ASTROS-ID, which allows analysts to "tune" or "update" models so that some of the results they predict (namely, selected natural frequencies and mode shapes) more closely match measured values. This report describes ASTROS-ID and some test cases that were used to exercise it.

> NOTE: At the time of writing (December 1991), this software had not been tested by anyone but its author. While it has been successfully exercised by the author, as reported below, there are likely to be bugs, as with all software. Furthermore, as discussed below, the software allows the user wide latitude in formulating problems, and insufficient research has been done to determine which approaches to use. Therefore, ASTROS-ID should only be used for research and not for "production" problems.

2. System Identification using ASTROS

In the context of structural analysis, "system identification," also called "model updating" or "model tuning," refers to a process whereby uncertain parameters of an analytical model are identified in a systematic manner. When test data are available for a structure that has been modeled, the identification process seeks values of these parameters that reduce discrepancies between certain measured responses values and those predicted by the model. A great deal of work has been done in this field; Ref. 1 may be consulted for a summary.

The purpose of this effort was to extend the work begun by Ewing and his colleagues at WL/FIBR concerning the use of mathematical optimization for system identification (Ref. 2). Their work studied the use of constrained or unconstrained minimization in which the independent variables are aspects of a finite element model that are uncertain and the objective is to modify those physical variables[1] so as to minimize discrepancies between measured and predicted natural frequencies. While their work was generally successful, they were hampered because their underlying analysis capability was a simple in-house finite element program limited to bars and rods in two dimensions. The obvious way to remove this limitatior to extend ASTROS, a multi-disciplinary design optimization and analysis ige developed for WL/FIBR, so that it could perform the same system identification function. Incorporating this capability into ASTROS has made it possible to take advantage of many valuable features available in that code from the point of view of the user as well as the developer. The user will be able to take advantages of a fairly complete element library, rigid elements, multi-point constraints, various eigenvalue methods, and manipulation of data in the ASTROS database through ICE (Ref. 3) or user-written Fortran programs. As explained in Section 4, the developer took advantage of ASTROS rich library of functional modules, its MAPOL language, the CADDB database, and the system generation feature.

2.1 Mathematical Formulation

The mathematical programming problem incorporated in ASTROS-ID is defined here. The complete objective function is:

$$
\begin{aligned}
f \; = \; & \sum_{j \in J_1} a_j \left(\frac{\lambda_j}{\bar{\lambda}_j} - 1 \right)^2 \\
& + \sum_{j \in J_2} b_j \left\| \Phi_j - \bar{\Phi}_j \right\|^2 \\
& + \sum_{\ell \in R} c_\ell \left(\frac{r_\ell}{r_\ell^0} - 1 \right)^2
\end{aligned}
$$

[1] "Design variable" is the corresponding term in design optimization. This term is used in the software source code and may appear in this report in some places.

$$+\frac{d}{M}\left(M - M_0 - \sum_{\ell=1}^{n} m_\ell\right)^2 \tag{1}$$

where:

J_1 The set of modal frequencies to be included in the objective
λ_j Computed eigenvalue for mode j
$\bar{\lambda}_j$ Measured eigenvalue for mode j
$\Phi_{j,k}$ Computed mode shape entry for mode j, DOF k
$\bar{\Phi}_{j,k}$ Measured mode shape entry for mode j, DOF k
$\| \Phi_j - \bar{\Phi}_j \|^2$ $\sum_{k \in K_j} (\Phi_{j,k} - \bar{\Phi}_{j,k})^2$ for mode j.
J_2 Set of reduced degrees of freedom
R The set of physical variables whose deviations from their initial values are to be included in the objective
r_ℓ Physical variable value
r_ℓ^0 Initial physical variable value
n Number of physical variables
M Measured total mass
m_ℓ Mass associated with physical variable ℓ
M_0 Total FEM mass not associated with any physical variable

a_j, b_j, c_ℓ, d Weighting coefficients
K_j The set of degrees of freedom that were measured for mode j. (Often this number will be the same for all j but distinct K_j per mode allows exclusion of bad data points.)

The complete set of constraint functions is as follows:

$$\left| \frac{\lambda_j}{\bar{\lambda}_j} - 1 \right| < e_j \text{ for } j \in J_3 \tag{2}$$

$$\left\| \Phi_j - \bar{\Phi}_j \right\|^2 < f_j \text{ for } j \in J_4 \tag{3}$$

where:

J_3 The set of frequencies to be constrained to match the measured frequencies within a given tolerance.
J_4 The set of mode shape entries to be constrained to match measured mode shapes within a given tolerance.
e_j, f_j Tolerances

Note: mode shapes are normalized so that:

$$\max_{k \in K_j} |\Phi_{j,k}| = 1 \; \forall \; j \in J_3$$

$$\max_{k \in K_j} |\bar{\Phi}_{j,k}| = 1 \; \forall \; j \in J_4$$

This formulation provides the user with many choices. A particular natural frequency value can be pursued either by including a term in the objective function (the a_j term) or by specifying a constraint (e_j tolerance). The same is true of mode shape entries (b_j and f_j). Another term is provided in the objective function to penalize large excursions in physical variable values (c_j). Where there might otherwise be multiple local minima in a particular formulation, this term might help guide the process toward the most "conservative" one, i.e., one corresponding to the most modest changes in physical variables. Finally, there is an optional term that influences the final "design" so that it matches the measured mass of the structure (d term)

2.2 Optimization Strategy

ASTROS-ID uses basically the same mathematical programming strategy as does ASTROS.[2] Operations are carried out in the following sequence:

1. Eigenvalue analysis.
2. Sensitivity analysis of eigenvalues and/or eigenvectors as required.
3. Formulation and solution of an approximate optimization problem.
4. Cross-orthogonality check to detect possible shifts in the ranking of modes.
5. If global convergence has occurred, perform a final eigenvalue analysis and exit.
6. Otherwise go to step 1.

These steps are elaborated below.

2.2.1 Eigenvalue Analysis

An ordinary eigenvalue analysis is performed. All options available to ASTROS users are available in ASTROS-ID, including Guyan or Generalized Dynamic Reduction, and of choice of Givens' method or the inverse power method. It is incumbent upon the user to perform a preliminary analysis to identify the modes that correspond to measured modes, presumably with the aid of finite-element graphics software, and to

[2] The current release of ASTROS provides an optimality criteria approach to optimization as an alternative to mathematical programming. ASTROS-ID does not support such an alternative.

ensure that all the required modes are computed at each iteration. If modes switch rank from one iteration to the next, ASTROS-ID will detect the shift, but it is still possible that a mode of interest may have increased in frequency or rank to the point where it is no longer computed in a particular iteration. To prevent this, the user should compute a few modes beyond the highest one of interest.

2.2.2 Sensitivity Analysis

ASTROS-ID forms lists of frequencies and eigenvectors whose sensitivity is required. Derivations of frequency and eigenvector sensitivities using Nelson's method may be found in Appendix A. Sensitivities of stiffness and mass matrices are formed by finite-difference operations. These sensitivities provide a starting point for the analytical computation of response sensitivities. Nelson's method has limitations which should be noted: (1) excessive computation costs where more than a few physical variables and/or mode shapes are involved, and (2) inability to compute sensitivities of equal or very closely spaced roots.

2.2.3 Approximate Problem

The "approximate problem" in ASTROS-ID is formulated and solved as follows:

First, frequencies and mode shape entries are extrapolated linearly from the starting point in design space. Extrapolation in terms of reciprocal values of design variables rather than direct values is sometimes advantageous and is, in fact, used in ASTROS wherever there is no linking. However, no attempt has been made to use reciprocal variables in ASTROS-ID.

Move limits are provided to limit the excursion of physical variables to regions of design space where the approximate frequencies and mode shapes are reasonably accurate. By default, excursions for each approximate optimization are limited to factors of 0.5 to 2.0.

Micro-DOT is a popular general-purpose optimization package. The version incorporated in ASTROS is used, with the well-known "0-5-7" option (i.e., the constrained nonlinear minimization algorithm known as "modified feasible directions"). No attention has been given to alternative methods within Micro-Dot. This is because efficiency and accuracy of the optimization method are not significant concerns when approximate models are used. Efficiency is not a problem because extrapolated frequency and mode shape values can be calculated in practically no time. Accuracy is not important because there is no need to find a precise minimum value of a function which is only an approximation of the true objective function.

2.2.4 Cross-orthogonality Check

Frequencies and mode shapes to be matched are identified by the user in terms of their rank in the ordered list of computed frequencies. This can pose a problem if modes change r ınk as the optimization proceeds. Thus, for example, if the user wanted to track the first bending mode, this mode might start out as the second mode but due to changes in physical variables, it might become the third mode at some subsequent iteration. If rank switching were not detected by the software, the optimization process could become hopelessly lost if, for example, it tried to make a computed torsion mode match a measured bending mode.

The remedy used in ASTROS-ID is a cross-orthogonality check between mode sets from consecutive iterations.[3] Thus, if the mode shape matrices from two successive iterations are $\Phi^{(n-1)}$ and $\Phi^{(n)}$, the following matrix triple product is performed:

$$\mathbf{C} = \Phi^{(n-1)T}\mathbf{M}\Phi^{(n)}$$

If there were no changes at all between the two iterations, C would be a diagonal matrix of modal mass values. If there were small changes, the matrix would be diagonally dominant. The premise of the method here is that changes will be small enough that one number will stand out in each column of C, and its row and column numbers will indicate the rank numbers of the corresponding modes from the two iterations. The algorithm is as follows:

1. Compute C as above.
2. Examine the columns of C successively:
   (a) Sort the column on absolute values.
   (b) Note the row position of the largest absolute value.
   (c) Tentatively identify the mode having that row number with the previous iteration's mode having the current column number.
   (d) If the new mode number has already been assigned, try the second-highest term in absolute value; proceed until an unassigned mode is found. It is not likely that this branch of the code will be exercised often, but it is included to avoid failure of the method.
3. Fill in a new record of an unstructured entity called MTRACE, a list of integers in which the positions are the starting mode numbers and the values are the current mode numbers.

The method will fail if a mode is modified beyond recognition, i.e., if there is no predominant term in a particular column of C. However, if this happens, it is difficult to envision any method that would not fail.

[3] This cross-orthogonality check should not be confused with the "modal assurance criterion" which is often used in modal testing.

2.3 The User Interface in ASTROS-ID

ASTROS-ID uses many of the existing ASTROS bulk data entries plus some new ones. The existing ASTROS entries that are currently recognized by ASTROS-ID are:

ASET	ASET1	CBAR	CELAS1	CELAS2
CIHEX1	CIHEX2	CIHEX3	CMASS1	CMASS2
CONLINK	CONM1	CONM2	CONROD	CONVERT
CORD1C	CORD1R	CORD1S	CORD2C	CORD2R
CORD2S	CQDMEM1	CQUAD4	CROD	CSHEAR
CTRIA3	CTRMEM	DCONALE	DESELM	DESVARP
DESVARS	DLAGS	DYNRED	EIGR	ELEMLIS
ELIST	EPOINT	GDVLIST	GPWG	GRAV
GRID	GRIDLIS	JSET	JSET1	MAT1
MAT2	MAT8	MAT9	MPC	MPCADD
OMIT	OMIT1	PBAR	PCOMP	PCOMP1
PCOMP2	PELAS	PIHEX	PLIST	PMASS
PQDMEM1	PROD	PSHEAR	PSHELL	PTRMEM
RBAR	RBE1	RBE2	RBE3	RROD
SAVE	SEQGP	SET1	SET2	SHAPE
SPC	SPC1	SPCADD	SPOINT	SUPORT

Descriptions of all these bulk data entries may be found in the ASTROS User's Manual (Ref. 4).

Four new bulk data entries have been added to ASTROS-ID. They are TFREQ (for specification of natural frequencies), TSHAPE (for entry of measured mode shapes at selected degrees of freedom), SYSID (for c and d coefficients), and DVCOEF (for the c_ℓ coefficients of Equation 1). These entries are described in detail in Appendix B.

For ASTROS-ID, the standard MAPOL solution sequence of ASTROS was removed and replaced by a new MAPOL sequence which supports only eigenvalue analysis. This approach was chosen instead of modification of the existing ASTROS sequence because the existing sequence is so complex, spanning many disciplines such as aerodynamics that are irrelevant to system identification. Modification of the long complex ASTROS sequence would have been more difficult than creation of a new sequence. The ASTROS-ID solution includes the following features:

- Provision for optimization or analysis only. This and other user selections appear in the "solution control packet" as in ASTROS.

- Elimination of single- and multi-point constraints.

- Guyan reduction.

- Generalized dynamic reduction.
- Choice of eigenvalue methods.
- Recovery of omitted degrees of freedom when a reduction method is used.
- Recovery of eigenvector components made dependent by multi-point constraints.

Later, addition of a static analysis capability is contemplated. This will be simple if no constraint deletion facility is required. In design optimization, users typically specify that no stress may exceed a specified value. Since most element stresses fall well below the allowable value, ASTROS and other design optimization codes temporarily delete stress constraints for elements whose stresses are well below allowables prior to each optimization cycle in order to avoid needless constraint computations in the optimization module. However, this should not be necessary in ASTROS-ID because users will specify only a limited number of stress values.

3. Review of the ASTROS Software Structure

ASTROS is built upon a number of well-designed software elements that facilitate extensions like ASTROS-ID (Ref. 5). While it would be an exaggeration to say that such extensions are "easy," it is safe to say that they are less difficult and more foolproof than extensions to NASTRAN, and vastly simpler than extensions to a monolithic Fortran code. The major elements are (1) the matrix manipulation language, MAPOL, (2) the relational database, CADDB, (3) the system generation process, SYSGEN, and (4) the interactive query system, ICE. These are all reviewed briefly here; the interested reader should consult the ASTROS manuals, especially the Programmer's Manual, for more details. Also, partial listings of the respective files that are used for generation of ASTROS-ID may be found in the Appendices. The listings include only those entries that are unique to ASTROS-ID.

3.1 MAPOL, the Matrix Language

The MAPOL language specifies the sequence of operations that ASTROS carries out. While its most important function is the specification of matrix operations, it also manipulates relational data and scalars. In addition to in-line operations, the MAPOL code calls functional modules which are coded in FORTRAN.

3.2 CADDB, the Relational Database

CADDB is a relational database system developed for ASTROS but suitable for engineering applications other than ASTROS. In accordance with the requirements of finite element analysis and optimization, CADDB supports three kinds of data "entities:" relations, matrices, and unstructured entities.

Relations may be viewed as tables. Each column in a table has a name (called an "attribute") and a data type (real, integer, or character). This descriptive information about a relation is called its "schema." CADDB provides convenient operations on relations, including "projections" (which may be thought of as requests to extract only specified columns from tables), conditions (requests to select only rows that satisfy certain conditions), selective updating, etc. When adding new capabilities to ASTROS, it is convenient to use relations to store information that contains mixtures of integer, real, and character data; especially information coming directly from bulk data cards.

Matrices are stored by columns, with transparent packing to reduce storage of zeroes. Columns may be accessed sequentially or directly. Matrices always contain "real" numbers, either single- or double-precision.

Unstructured entities are typically used for temporary storage of data that has no structure except records, which may vary in length.

3.3 The SYSGEN Process

The SYSGEN process is the most important attribute of ASTROS from the standpoint of the developer. SYSGEN is a software process that generates a revised ASTROS system from a number of text files that describe its operation. The ASTROS system consists of binary code plus a system database. That is, information about the operations of ASTROS is stored in a special CADDB database that is not accessed by the user, but only by the ASTROS binary code.

The five text files that drive the ASTROS system are described briefly below. More information is available in Sections 3.3 and 3.4 of the ASTROS Programmer's Manual.

3.3.1 Functional Module Definition

This file, called MODDEF.DAT on most computers, describes the modules that may be called by a MAPOL program. Each module name and calling sequence (list of argument types – relations, matrices, unstructured entities; real, integer and logical scalars) are listed, along with a few lines of FORTRAN code, which get linked with the ASTROS object code after SYSGEN finished.

3.3.2 Standard Solution Algorithm Definition

This file (usually MAPOLSEQ.DAT) is the MAPOL source code that is executed when ASTROS runs. However, users are free to alter the MAPOL code with EDIT commands, or to replace it with their own stand-alone code. The MAPOL code consists of a sequence of definitions of the data entities used by the MAPOL code (relations, matrices, unstructured entities; real, integer and logical scalars). The definition section includes not only the data entities that appear explicitly in the MAPOL code, but also "hidden" data entities that are accessed by functional modules but do not appear in their calling sequence. Although probably necessary in order to avoid unwieldy calling sequences, the use of hidden entities makes MAPOL less "clean" than it might otherwise be. This is because modules can communicate among themselves by means of hidden entities whereas the MAPOL code gives the impression that all communication is taking place through calling sequences.

3.3.3 Bulk Data Template

This file (usually TEMPLATE.DAT) contains "templates" of all the bulk data entries that ASTROS is expected to recognize. There are six logical lines of information for each bulk data type. If any of the logical lines exceeds 80 characters in length, another set of 6 physical lines appears immediately after the first 6. The six lines contain the following information:

1. Labels for each field.
2. Data types for each field (integer, real, character).
3. Default values for each field.
4. Error checks (e.g., numbers that must be positive).
5. Location in a temporary array into which each datum is to be loaded.
6. Names of the relation to receive the bulk data, and a list of the attributes which are to be loaded from the array.

3.3.4 Relational Schema Definition

This file (typically RELATION.DAT) contains the name of each relation followed by a list of its attribute names and types.

3.3.5 Error Message Text Definition

This file (ERRMSG.DAT) contains error message text, which may be accessed by calls from the functional modules.

3.4 ICE: interactive access to CADDB

ICE ("Interactive CADDB Environment") is an interactive software system that allows users to query and update CADDB databases. It is a useful alternative to examining "print" files generated by ASTROS. A sophisticated query language allows users to examine data using various qualifiers. For example, ASTROS stores design variable information in a relation called GLBDES. To examine the design history of design variable number 101, one may simply type:

```
SELECT NITER,DVID,VALUE FROM GLBDES WHERE DVID=101;
```

or to examine the history of the first three frequencies:

```
SELECT NITER,MODENO,CFREQ FROM LAMBDA WHERE MODENO<4;
```

ICE is described fully in Ref. 3.

4. Implementation

ASTROS-ID was implemented as a special version of ASTROS based on a special version of the SYSGEN files. For the most part, these files were created by modifying the existing ASTROS files. The MAPOL sequence is a complete stand-alone sequence rather than a modification of the existing multidisciplinary sequence. It was felt that this would be simpler than modifying the ASTROS sequence, which is very complex due to the multiplicity of disciplines that are supported, most of which are irrelevant to system identification. The ASTROS-ID sequence supports only normal modes analysis and optimization for system identification relative to frequencies and mode shape entries. At a later date, static analysis may be added. All the features available with normal modes in ASTROS are available with ASTROS-ID, including a choice of methods (inverse power or Givens) and a choice of reduction methods (Guyan, generalized reduction, or none).

5. Sample Problems

For this effort, there has been no attempt to tune a finite element model to actual test data. At this stage, we wanted to concentrate on developing and debugging the software and avoid all the questions raised by real test data, interesting as these questions may be. Instead, we performed a number of "computer experiments" in which a model was deliberately mistuned and then given to ASTROS-ID to see whether it could recover the original design variable values.

5.1 AMRAAM Beam Model

The first problem that was attempted was the AMRAAM beam model reported by Ewing in his paper (Ref. 2). This model consists of 27 beam elements representing the main body of the missile plus two beam elements representing mounts, as shown in Figure 1. The elements were grouped as follows:

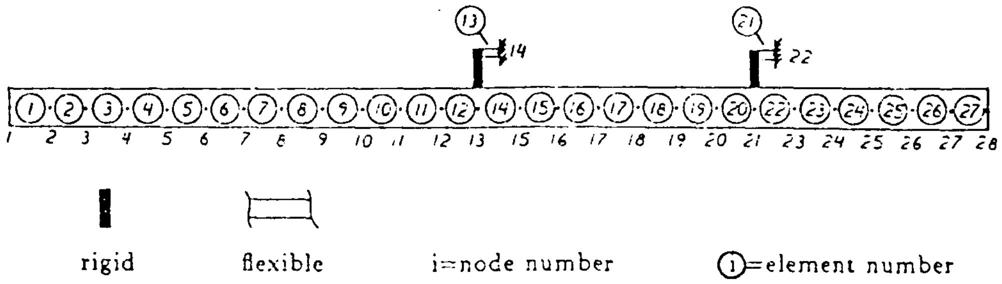

Figure 1: AMRAAM Beam Model

- Elements on the left end (1-12): PBAR 1; DESVARP 1
- Elements in the center (14-20): PBAR 2; DESVARP 2
- Elements on the right end (22-27): PBAR 3; DESVARP 3

Of course, if this were a real system identification problem, the two elements representing mount springs would probably be more uncertain and, therefore, better candidates for physical variables. For the purpose of an academic "computer experiment," however, the main beam elements were more convenient. A number of variations on this problem were attempted. Results may be seen in Tables 1 through 4.

Table 1: AMRAAM beam, first run.

Problem 1a: AMRAAM beam with three D.V., one Freq.

Design variables:

ID	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	PBAR,1	2.0	2.4	2.467		
2	PBAR,2	2.0	1.6	1.734		
3	PBAR,3	2.0	2.0	2.000		

Frequencies included in the objective function:

Mode	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	First bending	2.293	2.081	2.291		

No constraints.

Remark: This problem is clearly underdetermined (i.e., with three variables, there are multiple solutions that yield the correct first frequency). Add another frequency to the objective function.

Table 2: AMRAAM beam, second run.

Problem 1b: AMRAAM beam with three D.V., two Freq.

Design variables:

ID	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	PBAR,1	2.0	2.4	2.216	2.271	
2	PBAR,2	2.0	1.6	1.789	1.803	
3	PBAR,3	2.0	2.0	2.406	2.372	

Frequencies included in the objective function:

Mode	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	First bending	2.293	2.081	2.245	2.292	
2	Second bending	8.000	6.725	7.949	7.993	

No constraints.

Remark: This problem is still underdetermined. Add a third frequency to the objective function.

Table 3: AMRAAM beam, third run.

Problem 1c: AMRAAM beam with three D.V., three Freq.

Design variables:

ID	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	PBAR,1	2.0	2.4	2.045	1.991	
2	PBAR,2	2.0	1.6	1.827	1.996	
3	PBAR,3	2.0	2.0	2.324	2.316	

Frequencies included in the objective function:

Mode	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	First bending	2.293	2.081	2.178	2.276	
2	Second bending	8.000	6.726	7.948	7.621	
3	Third bending	15.72	18.49	15.85	15.62	

No constraints.

Remark: This problem now works well, although all frequencies appear to be insensitive to the third design variable. Now try one frequency and three entries from the corresponding mode shape.

Table 4: AMRAAM beam, fourth run.

Problem 1d: AMRAAM beam with three D.V., one Freq., three entries of one mode shape

Design variables:

ID	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	PBAR,1	2.0	2.4	1.719	1.991	
2	PBAR,2	2.0	1.6	1.899	1.996	
3	PBAR,3	2.0	2.0	2.316	2.315	

Frequencies included in the objective function:

Mode	Description	Base	Start	Iter.1	Iter.2	Iter.3
1	First bending	2.293	2.081	1.953	2.281	

Mode shape entries included in the objective function:

Mode	DOF	Base	Start	Iter.1	Iter.2	Iter.3
1	1,Z	-0.6887	-0.654	-0.698	-0.6889	
1	17,Z	+0.0307	+0.0499	+0.0258	+0.0305	
1	28,Z	-0.0697	-0.1155	-0.0573	-0.0684	

No constraints.

Remark: Mode 1 eigenvalue & vector both insensitive to DV 3. Excursion terms in the objective should limit changes in such DV.

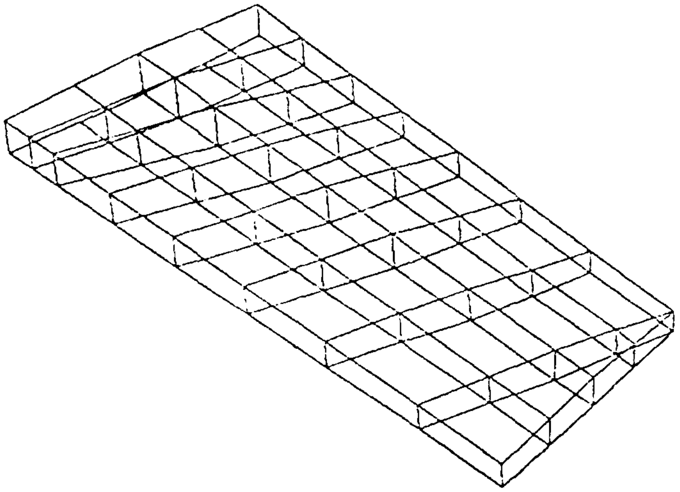

Figure 2: Intermediate wing box

5.2 Intermediate Wing Box

The wing box model shown in Figure 2 has been used as a sample problem for optimization with ASTROS. Here the problem has been changed slightly. Instead of design optimization, we are trying to identify the fuselage compliance at the root of the wing by inserting spring elements there, and selecting their values to match the first few natural frequencies. Instead of hard constraints at the root of the wing, springs were added in the outboard direction to simulate fuselage compliance. Among the five pairs of nodes at the root, the inboard and outboard pairs were assigned spring rates of 5×10^6 lb/in per node, and the others 2×10^6. Eight normal modes were computed using these values. The spring rates were then deliberately mistuned to see if the original values could be recovered. Again, this problem is simply a "computer experiment," i.e., no real test data were involved. However, the use of artificial springs at the root makes this problem somewhat more representative of a real system identification problem.

5.2.1 Possible Local Minimum

In anticipation of possible problems with relative minima, a contour plot was prepared to see how well-behaved the objective function appeared for this problem. This plot may be seen in Figure 3. In this plot, all data were derived from "exact" eigenvalue analyses. The plot shown in Figure 4, by contrast, is derived from an approximate model where the exact analysis was conducted at the point (1.0, 1.0). As expected, the approximate objective function deviates substantially from the exact function. Nevertheless, ASTROS-ID was able to solve this problem easily in a few iterations, as the following paragraphs show. On completion of each approximate optimization,

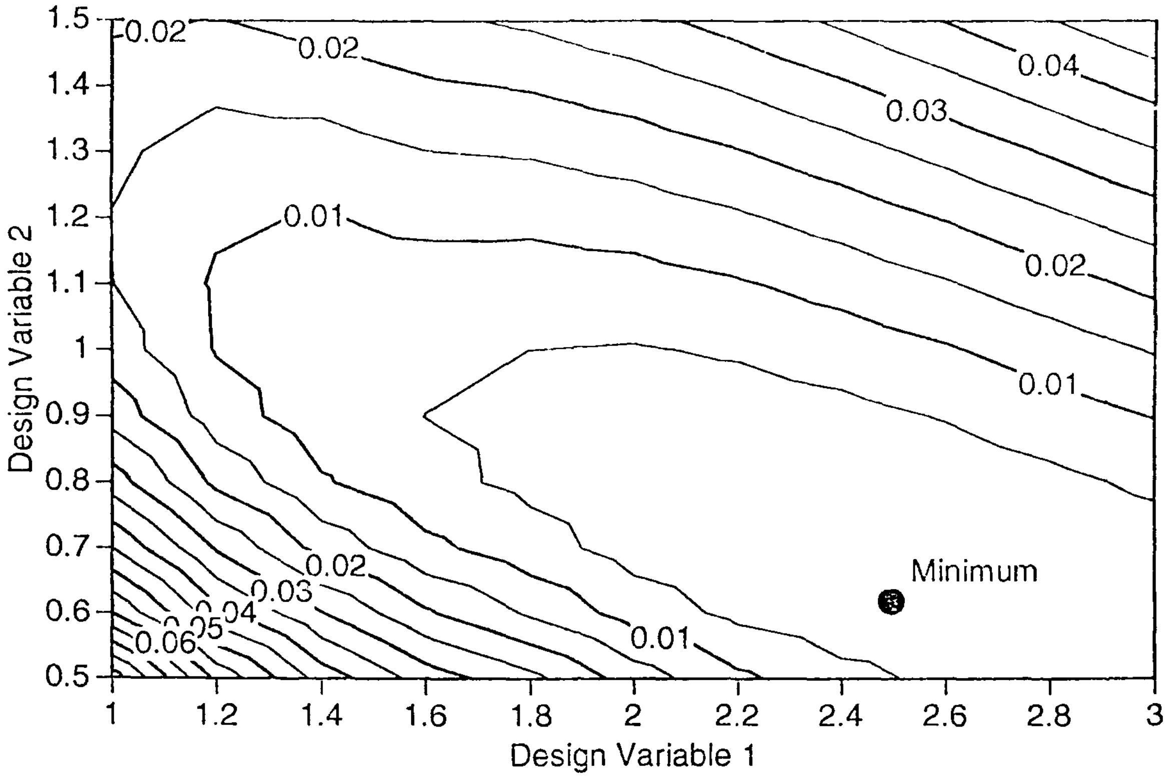

Figure 3: Contour plot of the "exact" objective function for the intermediate wing.

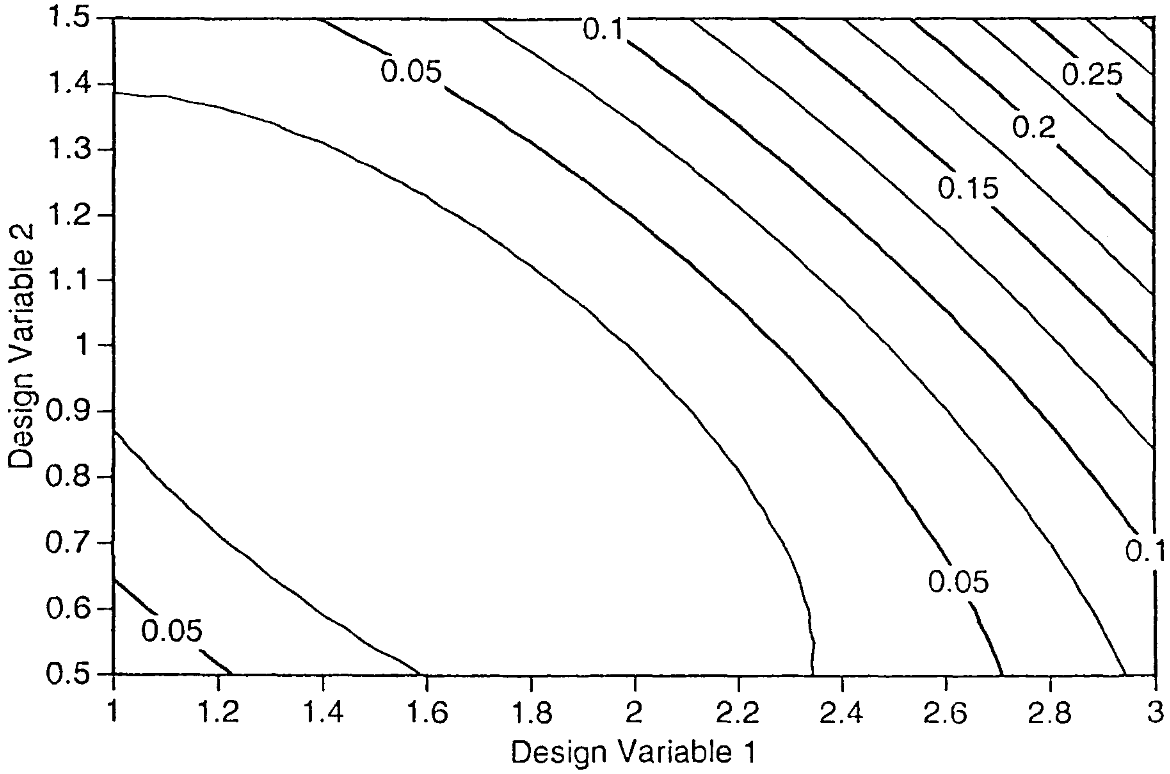

Figure 4: Contour plot of the approximate objective function for the intermediate wing.

a new "exact" analysis is conducted at the point which was determined to be the optimum of the approximate problem. Then a new approximate problem is solved and the process continues until convergence is achieved.

Table 5: Design history for the intermediate wing, using the objective function.

Iteration	DV-1	DV-2
0	0.2	4.0
1	0.4	2.0
2	0.640	1.266
3	0.680	1.295
4	0.683	1.291
5	0.940	1.023
Alternate starting point:		
0	10.00	0.100
1	5.000	0.100
2	2.500	0.100
3	2.055	0.200
4	1.656	0.400
5	1.067	0.800
6	1.026	0.961
7	1.013	0.976
8	1.006	0.989
9	1.002	0.996

5.2.2 Results using objective function terms

Table 5 shows the design variable history in scaled terms (i.e., these are the values of the VALUE attribute of the DESVAR relation, whose initial values appear on the DESVAR bulk data entry). Two alternate starting points are selected, the second quite far from the optimum. As the results indicate, convergence to the correct values (1.0, 1.0) was obtained in each case without any trouble.

5.2.3 Results using constraints

The problem was reformulated using constraints on the first seven frequencies and no objective function. The results are as shown in Table 6.

The problem was defined with the constraint parameter e_j of Eq. (2) set to 1×10^6. Although the process was apparently heading for the correct answer (1.0, 1.0) it stopped after the third iteration because all constraints were satisfied to within the "constraint tolerance" parameter (CTL). That is, constraints functions that are

Table 6: Design history for the intermediate wing, using constraints.

Iteration	DV-1	DV-2
0	10.0	0.1
1	5.0	0.1
2	2.5	0.1
3	1.79	0.2

supposed to be negative are considered inactive if their value does not exceed a positive number given by CTL. It should be possible to adjust this parameter within Micro-DOT so that constraints are always considered active when they are negative.

In actual practice, it will probably be wise to include the most important frequencies in the objective function and to control less important frequencies using constraints.

6. Conclusions and Extensions

A capability for system identification using mathematical programming has been developed based on ASTROS. ASTROS-ID provides for identification of physical variables by matching natural frequencies and/or mode shape entries to measured values. All the conveniences of ASTROS with regard to generality of the finite element model, eigensolution techniques, and "physical variable" choices (i.e., design variables) are available in ASTROS-ID.

6.1 Questions Remaining to be Addressed

While this effort was successful in providing the requested capability (subject to possible minor bugs), many interesting questions remain. These include (1) whether constraints or terms in the objective function are more effective, (2) how to choose weighting coefficients, (3) how to avoid underdetermined formulations, (4) how to choose physical variables, and (5) the effects of tuning on modes not included in the tuning process. It is hoped that ASTROS-ID will provide a good testbed for investigation of these questions.

6.2 Caveats and Limitations

As was mentioned, the really interesting questions regarding the use of optimization for system identification concern the choice of problem formulation by the user. ASTROS-ID allows wide latitude in this regard. If given an underdetermined problem formulation, for example, it will blindly find the first local minimum that it finds and exit. Choices of weighting coefficients are left to the user and may influence the outcome significantly. It is very important that users bear these facts in mind and treat ASTROS-ID solely as a research tool at its present stage.

Nelson's method for eigenvalue sensitivity analysis is rather time-consuming. It requires one matrix decomposition for each eigenvector whose sensitivity is required, following by one forward/backward substitution for each design variable. Thus either a large number of eigenvector sensitivities or a large number of physical variables may lead to excessive computing time.

As was mentioned, Nelson's method will fail when an attempt is made to calculate sensitivities of repeated roots (i.e., normal modes that have exactly equal or very near equal frequencies). A modification to Nelson's method that deals with repeated roots has been published in the literature and may be added to ASTROS-ID in the future.

A tracking capability for handling mode rank switches has been implemented but has not been tested as of this writing. Objective function terms designed to match analytical mass to measured mass have not been exercised, nor have excursion terms, which are intended to find a solution that deviates minimally from starting values.

Many combinations of element types, constraints, and reduction methods are possible. Not all of these have been tested, although no problems are anticipated.

Finally, not much attention has been paid to error messages associated with the new featu.es of ASTROS-ID. These may receive more attention in future work.

6.3 Recommendations for further work

The most obvious need at this time is to exercise the code, both to shake out possible bugs or shortcomings, and to gain insight into the advantages and disadvantages of the various options that are open to the user. The mode switching code should be checked out. A problem needs to be run using live test data.

Possible extensions include:

- Incorporation of static analysis and identification using static displacements and/or stresses. This would be integrated with the initial dynamic capability.
- Improvement or replacement of Nelson's method to handle repeated roots and to increase efficiency.
- Inclusion of Bayesian estimation as an alternative to mathematical programming.
- Investigation of second-order frequency sensitivities. It can be shown that if mode shape sensitivities have been computed, they can be used to compute second-order sensitivities of natural frequencies, which in turn could be used to form more accurate approximate models.

7. References

1. Imregun, M., and Visser, W. J., "A Review of Model Updating Techniques," *Shock and Vibration Digest*, Vol. 13, Jan 1991, pp 9-20.

2. Ewing, M. S. and Kolonay, R. M., "Dynamic Structural Model Modification using Mathematical Optimization Techniques," Opti 91, Boston, June 1991.

3. Herendeen, D. L. and Ludwig, M. R., "Interactive Computer Automated Design Database (CADDB) Environment User's Manual," AFWAL-TR-88-3060.

4. Neill, D. J., Johnson, E. H., and Herendeen, D. L., "Automated Structural Optimization System (ASTROS)," Vol. II – User's Manual, AFWAL-TR-88-3028.

5. Neill, D. J., Johnson, E. H., and Hoesly, R. L., "Automated Structural Optimization System (ASTROS)," Vol. IV – Programmer's Manual, AFWAL-TR-88-3028.

Appendix A: Nelson's Method for Eigenvector Sensitivity Analysis in NASTRAN or ASTROS

The following pages describe the bulk data entries that are unique to ASTROS-ID. All of ASTROS's general rules about bulk data entries, as described in Ref. 4, apply to these entries.

Nelson's method for eigenvector sensitivity in NASTRAN or ASTROS

In this derivation, boldface symbols indicate matrices or vectors. A "hat" ($\hat{}$) over a symbol indicates differentiation with respect to some design variable. Subscripts indicate set membership and superscripts are mode numbers. Set notation is as follows:

G	all DOF
M	constrained by MPC's
N	not constrained by MPC's
S	constrained by SPC's
F	not constrained by SPC's

Note that $G = M \cup N$ and $N = F \cup S$.

Assume that stiffness and mass matrix sensitivities $\hat{\mathbf{K}}_{GG}$ and $\hat{\mathbf{M}}_{GG}$ are available, as well as their partitions in the various sets. We are solving the eigenvalue equation:

$$[\mathbf{K}_{GG} - \lambda_i \mathbf{M}_{GG}]\, \Phi_G = 0 \tag{4}$$

for mode i. Differentiate and arrange terms:

$$[\mathbf{K}_{GG} - \lambda_i \mathbf{M}_{GG}]\, \hat{\Phi}^i_G = \left[\hat{\lambda}_i \mathbf{M}_{GG} + \lambda_i \hat{\mathbf{M}}_{GG} - \hat{\mathbf{K}}_{GG}\right] \Phi^i_G \tag{5}$$

Premultiply by Φ^{iT}_G. The left-hand side then vanishes, and we can solve for $\hat{\lambda}_i$:

$$\hat{\lambda}_i = \frac{1}{m_i}\left[\Phi^{iT}_G \hat{\mathbf{K}}_{GG} \Phi^i_G - \lambda_i \Phi^{iT}_G \hat{\mathbf{M}}_{GG} \Phi^i_G\right] \tag{6}$$

where m_i is the modal mass $\Phi^{iT}_G \mathbf{M}_{GG} \Phi^i_G$. Proceeding with the derivation of eigenvector derivatives, let:

$$\begin{aligned} \mathbf{D}^i_{GG} &= \mathbf{K}_{GG} - \lambda_i \mathbf{M}_{GG} \\ \mathbf{F}^i_G &= \left[\hat{\lambda}_i \mathbf{M}_{GG} + \lambda_i \hat{\mathbf{M}}_{GG} - \hat{\mathbf{K}}_{GG}\right] \Phi^i_G \end{aligned}$$

so that Eq. (5) becomes:

$$\mathbf{D}^i_{GG} \hat{\Phi}^i_G = \mathbf{F}^i_G \tag{7}$$

We want the sensitivity expressions to obey MPC and SPC constraints, if present. That way any extrapolations in an approximate model will produce trial vectors that satisfy these constraints. If MPC's are present, partition $\mathbf{D}^i_{GG}$ and $\mathbf{F}^i_G$:

$$\begin{aligned} \mathbf{D}^i_{GG} &\Rightarrow \begin{bmatrix} \bar{\mathbf{D}}^i_{NN} & \mathbf{D}^i_{NM} \\ \mathbf{D}^i_{MN} & \mathbf{D}^i_{MM} \end{bmatrix} \\ \mathbf{F}^i_G &\Rightarrow \begin{Bmatrix} \bar{\mathbf{F}}^i_N \\ \mathbf{F}^i_M \end{Bmatrix} \end{aligned} \tag{8}$$

and carry out the standard reductions:

$$\begin{aligned} \mathbf{D}^{i}_{NN} &\Leftarrow \bar{\mathbf{D}}^{i}_{NN} + \mathbf{D}^{i}_{NM}\mathbf{G}_{MN} + \mathbf{G}^{T}_{MN}\mathbf{D}^{i}_{MN} + \mathbf{G}^{T}_{MN}\mathbf{D}^{i}_{MM}\mathbf{G}_{MN} \\ \mathbf{F}^{i}_{N} &\Leftarrow \bar{\mathbf{F}}^{i}_{N} + \mathbf{G}^{T}_{MN}\mathbf{F}^{i}_{M} \end{aligned} \tag{9}$$

If there are no MPC's:

$$\begin{aligned} \mathbf{D}^{i}_{NN} &\Leftarrow \mathbf{D}^{i}_{GG} \\ \mathbf{F}^{i}_{N} &\Leftarrow \mathbf{F}^{i}_{G} \end{aligned} \tag{10}$$

If there are SPC's, do:

$$\begin{aligned} \mathbf{D}^{i}_{NN} &\Rightarrow \begin{bmatrix} \mathbf{D}^{i}_{FF} & \mathbf{D}^{i}_{FS} \\ \mathbf{D}^{i}_{SF} & \mathbf{D}^{i}_{SS} \end{bmatrix} \\ \mathbf{F}^{i}_{N} &\Rightarrow \begin{Bmatrix} \mathbf{F}^{i}_{F} \\ \mathbf{F}^{i}_{S} \end{Bmatrix} \end{aligned} \tag{11}$$

otherwise:

$$\begin{aligned} \mathbf{D}^{i}_{FF} &\Leftarrow \mathbf{D}^{i}_{NN} \\ \mathbf{F}^{i}_{F} &\Leftarrow \mathbf{F}^{i}_{N} \end{aligned} \tag{12}$$

The equation to be solved is now:

$$\mathbf{D}^{i}_{FF}\hat{\Phi}^{i}_{F} = \mathbf{F}^{i}_{F} \tag{13}$$

The matrix $\mathbf{D}^{i}_{FF}$ is singular. This difficulty may be resolved by expanding the eigenvector sensitivity for the F-set in terms of the original eigenvectors in the same set, i.e.:

$$\begin{aligned} \hat{\Phi}^{i}_{F} &= \sum_{k=1}^{F} \xi_k \Phi^{k}_{F} \\ &= C_i \Phi^{i}_{F} + \sum_{k=1}^{F} \tilde{C}_k \Phi^{k}_{F} \\ &= C_i \Phi^{i}_{F} + \mathbf{V}^{i}_{F} \end{aligned} \tag{14}$$

where:

$$\begin{aligned} \tilde{C}_k &= \xi_k \text{ when } k \neq i \\ &= \xi_i - C_i \text{ when } k = i \end{aligned} \tag{15}$$

Because Φ^{i}_{F} is the solution of the homogeneous equation $\mathbf{D}^{i}_{FF}\Phi^{i}_{F} = 0$, Eq. (13) becomes:

$$\mathbf{D}^{i}_{FF}\mathbf{V}^{i}_{F} = \mathbf{F}^{i}_{F} \tag{16}$$

Since $\mathbf{D}^i_{FF}$ is of rank $F-1$, Eq. (16) cannot be solved uniquely. We can arbitrarily set $\mathbf{V}^i_k = 0$, eliminate row and column k from $\mathbf{D}^i_{FF}$ and $\mathbf{F}^i_F$, and use the reduced form of Eq. (16) to solve for the other components of $\bar{\mathbf{V}}^i_F$:

$$\bar{\mathbf{D}}^i_{FF}\bar{\mathbf{V}}^i_F = \mathbf{F}^i_F \tag{17}$$

The determination of the coefficient C_i in Eq. (14) requires an extra equation, namely the normalization equation. Assume normalization to unit modal mass:

$$m_i = \Phi^{iT}_F \mathbf{M}_{FF} \Phi^i_F = 1 \tag{18}$$

Differentiating:

$$2\Phi^{iT}_F \mathbf{M}_{FF} \hat{\Phi}^i_F + \Phi^{iT}_F \hat{\mathbf{M}}_{FF} \Phi^i_F = 0 \tag{19}$$

Substituting Eq. (14) and solving for C_i:

$$C_i = -\frac{1}{m_i}\left(\frac{1}{2}\Phi^{iT}_F \hat{\mathbf{M}}_{FF} \Phi^i_F + \Phi^{iT}_F \mathbf{M}_{FF} \mathbf{V}^i_F\right) \tag{20}$$

Substituting C_i and $\mathbf{V}^i_F$ into Eq. (14) completes the computation of $\hat{\Phi}^i_F$. This vector must be expanded back to the G-set in the conventional way. If there are SPC's, do:

$$\hat{\Phi}^i_N \Leftarrow \left\{ \begin{array}{c} \hat{\Phi}^i_F \\ 0 \end{array} \right\} \tag{21}$$

otherwise:

$$\hat{\Phi}^i_N \Leftarrow \hat{\Phi}^i_F \tag{22}$$

If there are MPC's, do:

$$\begin{array}{rcl} \hat{\Phi}^i_M & \Leftarrow & \mathbf{G}_{MN}\hat{\Phi}^i_N \\ \hat{\Phi}^i_G & \Leftarrow & \left\{ \begin{array}{c} \hat{\Phi}^i_M \\ \hat{\Phi}^i_N \end{array} \right\} \end{array} \tag{23}$$

otherwise:

$$\hat{\Phi}^i_G \Leftarrow \hat{\Phi}^i_N \tag{24}$$

Appendix B: Bulk Data Entries

The following pages describe the bulk data entries that are unique to ASTROS-1D. All of ASTROS's general rules about bulk data entries, as described in Ref. 4, apply to these entries.

As mentioned elsewhere, the following ASTROS bulk data entries are also supported by ASTROS-1D.

ASET	ASET1	CBAR	CELAS1	CELAS2
CIHEX1	CIHEX2	CIHEX3	CMASS1	CMASS2
CONLINK	CONM1	CONM2	CONROD	CONVERT
CORD1C	CORD1R	CORD1S	CORD2C	CORD2R
CORD2S	CQDMEM1	CQUAD4	CROD	CSHEAR
CTRIA3	CTRMEM	DCONALE	DESELM	DESVARP
DESVARS	DLAGS	DYNRED	EIGR	ELEMLIS
ELIST	EPOINT	GDVLIST	GPWG	GRAV
GRID	GRIDLIS	JSET	JSET1	MAT1
MAT2	MAT8	MAT9	MPC	MPCADD
OMIT	OMIT1	PBAR	PCOMP	PCOMP1
PCOMP2	PELAS	PIHEX	PLIST	PMASS
PQDMEM1	PROD	PSHEAR	PSHELL	PTRMEM
RBAR	RBE1	RBE2	RBE3	RROD
SAVE	SEQGP	SET1	SET2	SHAPE
SPC	SPC1	SPCADD	SPOINT	SUPORT

Consult Ref. 4 for documentation on these entries.

Input Data Entry **TFREQ** Natural frequencies to be identified.

Description: Specifies natural frequencies to be included either in the objective function or in constraints.

Format and example:

TFREQ	SETID	MODE	FREQ	A	E	.	.	.
TFREQ	1	3	7.02		0.02			

Field	**Contents**
SETID	Not used at present (integer)
MODE	Mode number (integer > 0)
FREQ	Measured natural frequency in Hz (real > 0.0)
A	Coefficient for inclusion in the objective function (real, blank or > 0.0)
E	Constraint tolerance for inclusion as a constraint

Remarks:

1. Refer to the definition of the objective function and constraints in the main body of the report for the meanings of A and E. Either A or E should be included, but not both.

2. It is incumbent upon the user to run an initial analysis to correlate measured with analytical modes in order to identify the value to use for MODE.

Input Data Entry TSHAPE – Mode shape entries to be identified.

Description: Specifies mode shape entries to be included either in the objective function or in constraints.

Format and example:

TSHAPE	SETID	MODE	GRID	COMP	MEAS	B	F	
TSHAPE	1	3	101	3	0.02	1.0		

Field	**Contents**
SETID	Not used at present (integer)
MODE	Mode number (integer > 0)
GRID	GRID point ID (integer > 0)
COMP	displacement component (integer, 1 to 6)
MEAS	measured value (real $\neq 0$)
B	Coefficient for inclusion in the objective function (real, blank or > 0.0)
F	Constraint tolerance for inclusion as a constraint

Remarks:

1. The user must ensure that the displacement coordinate system at the indicated grid point matches the coordinate system used in the test, and that the indicated component matches the positive direction specified in the test. While rotation values may be used, it is very unlikely that such values will be available from tests, so COMP will typically be 1, 2, or 3.

2. At least two mode shape entries must be specified for each measured mode shape.

3. If optimization is to be performed, at least one TFREQ or TSHAPE entry must be present.

4. Refer to the definition of the objective function and constraints for the meanings of A and E. Either A or E should be included, but not both.

5. It is incumbent upon the user to run an initial analysis to correlate measured with analytical modes in order to identify the value to use for MODE.

Input Data Entry **SYSID** -- additional objective function entries.

Description: Specifics additional terms to be included in the objective function: physical variable excursion penalties and/or a mass identification term.

Format and example:

SYSID	SETID	DV	MASS	D	.	.	.	.
SYSID	1	1	17.3	1.0				

Field	**Contents**
SETID	Not used at present (integer)
DV	ID of a DVCOEF entry identifying physical variables whose excursions are to be penalized in the objective function (integer > 0 or blank)
MASS	Measured total mass (real > 0.0 or blank)
D	Coefficient the normalized mass penalty term (real > 0.0 or blank

Remarks:

1. Refer to the definition of the objective function in the body of the report for explanations of the excursion terms and mass terms in the objective function.

2. If a mass term is included, proper units must be used. That is, if a factor of $1/g$ has been entered on a CONVERT,MASS entry, then weight units should be used; otherwise mass units.

Input Data Entry **DVCOEF** - Physical variable excursion terms.

Description: Lists physical variables ("design variables") and coefficients whose excursions from starting values are to be included as terms in the objective function.

Format and example:

DVCOEF	SETID	DVID	C	DVID	C	DVID	C	.
DVCOEF	1	101	1.0	102	1.0			

Field	**Contents**
SETID	Set ID referenced by field 5 of the SYSID entry (integer > 0)
DVID	ID of an ASTROS design variable (integer > 0)
C	Coefficient (real > 0.0)

Remark: Refer to the definition of the objective funciton in the body of the report for an explanation of the excursion terms in the objective function.

Appendix C: List of New Modules in ASTROS-ID

The following is a list of modules that were written specifically for ASTROS-ID.

Module ACTLIST

Purpose: to identify eigenvalues and eigenvectors whose sensitivities are required. This done by scanning user TFREQ and TSHAPE entries. Arguments:

1. NITER (integer, input) iteration number.
2. LAMLIST (relation, output) list of eigenvalues whose sensitivity is required.
3. NUMLSENS (integer, output) length of LAMLIST.
4. PHILIST (relation, output) list of eigenvectors whose sensitivity is required.
5. NUMPSENS (integer, output) length of PHILIST.

Modules DVSAVE and DVREST

Purpose: To save and restore the current design variable values from/to relation GLBDES. Arguments:

1. NITER (integer, input) iteration number.
2. NDV (integer, input) number of design variables.

Module GETEVAL

Purpose: to get the current value of a particular eigenvalue.
Arguments:

1. LAMBDA (relation, input) eigenvalue table.
2. LAMLIST (relation, input) list of eigenvalues whose sensitivity is required.
3. LAMINDX (integer, input) index into LAMLIST.
4. MODENUM (integer, output) mode number from LAMLIST.
5. EVAL (real, output) eigenvalue from LAMBDA.

Module GETCOL

Purpose: to get a particular column from a matrix.
Arguments:

1. MAT (matrix, input) matrix from which to get a column.

2. ICOL (integer, input) column number.

3. VEC (matrix, output) vector containing the designated column.

Module GETSCAL

(Note: this module is called as a function.)

Purpose: to get a scalar value from a matrix.

Arguments:

1. MAT (matrix, input) matrix from which to get a scalar.

2. ICOL (integer, input) column number.

3. IROW (integer, input) row number.

Module INCRDV

Purpose: to increment design variable values in relation GLBDES for the purpose of finite-difference stiffness and mass sensitivity calculations.

Arguments:

1. IDV (integer, input) design variable number.

2. NDV (integer, input) number of design variables.

3. NITER (integer, input) iteration number.

4. DELTA (real, input) delta value to add to the specified d.v.

5. DVI (real, output) design variable increment.

Module MACTEST

Purpose: to check for rank switches among normal modes by testing the "modal assurance" matrix. A new record is appended to the unstructured entity MTRACE giving the current mode ranking.

Arguments:

1. NITER (integer, input) iteration number.

2. MAC (matrix, input) "modal assurance" matrix.

Module NELSON

Purpose: to compute the largest entry in a mode shape PHIA and to create a partitioning vector to eliminate that row and column. This is one of the steps in Nelson's method. The other steps are carried out by MAPOL code.

Arguments:

1. PHIA (matrix, input) mode shape matrix.
2. MODE (integer, input) column number of PHIA.
3. PARVEC (matrix, output) partitioning vector.
4. ISTAT (integer, output) return status

Module PUTSCAL

Purpose: to insert a scalar value into a matrix

Arguments:

1. KEY (integer, input) zero: initialize the matrix; plus one: insert a scalar into a particular location, minus one: insert a scalar into every
2. ICOL (integer, input)
3. ???

Module TRIPLE

Purpose: To perform a triple product of a vector times a matrix times a vector, as required for eigenvalue sensitivity.

Arguments:

1. VEC1 (matrix, input) left-hand vector.
2. MAT (matrix, input) matrix.
3. VEC2 (matrix, input) right-hand vector.
4. RESULT (real, output) result.

Module TUNEUP

Purpose: To perform one cycle of approximate optimization (this is the counterpart of the DESIGN module in ASTROS)

Arguments:

1. NITER (integer, input) iteration number.
2. NDV (integer, input) number of design variables.
3. MOVLIM (real, input) move limit.
4. CNVRGLIM (real, input) global convergence criterion.
5. GLBCNVRG (logical, output) convergence test result.

Appendix D: List of New Database Entities in ASTROS-ID

The following is a list of relations and matrices that were created for ASTROS-ID.

Relations TFREQ, TSHAPE, SYSID, DVCOEF

Storage for bulk data entries of the same names.

Relation LAMLIST

List of eigenvalues whose sensitivity is required.

Relation PHILIST

List of eigenvectors whose sensitivity is required.

Matrices KGGI and MGGI

Incremented stiffness and mass matrices, for finite-difference sensitivity calculations.

Subscripted matrices DKGG and DMGG

Finite-difference stiffness and mass sensitivities. Subscript values correspond to design variable numbers.

Matrix DD

Matrix used in frequency sensitivity calculations.

Matrices EVECF and EVECG

Temporary storage for individual columns of the mode shape matrices PHIF and PHIG respectively.

Matrix DLAM

Matrix of eigenvector sensitivities.

Matrix PARVEC

Partitioning vector used in Nelson's method.

Matrices DGG, DNN, DMM, DFF, DFF1

Matrices used in Nelson's method, relative to various sets (G, N, M, F, and F reduced by one).

Matrices FG, FF, FF1, FN, FN1, FM

Right-hand side vectors used in Nelson's method, relative to various sets.

VG, VN, VM, VF, VF1

Vectors used in Nelson's method.

LM1, LM2, LMK, LFF1

Vectors used in Nelson's method.

Matrix LAMMAT

Matrix version of the eigenvalue table LAMBDA.

Matrix DEVEC

Eigenvector sensitivity matrix. Row numbers are g-set degree of freedom numbers. Column numbering is as follows: first, columns for each design variable for the first mode, then another set of columns for all design variables for the second mode, etc.

Matrix DDEVEC

Temporary storage for eigenvector sensitivities.

Matrix MAC

"Modal assurance criterion" matrix measuring correspondence between mode shapes for successive iteration.

Matrix MACTEMP

Temporary storage for calculation of MAC.

Unstructured SAVEDV

Temporary storage for design variable v ılues

U.S. GOVERNMENT PRINTING OFFICE 750-113